

JON-0559US
K. KUWABARA
P/1905-16

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 2 月 1 8 日
Date of Application:

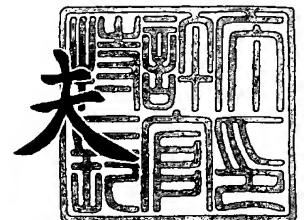
出 願 番 号 特 願 2 0 0 3 - 0 3 9 0 5 9
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 0 3 9 0 5 9]

出 願 人
Applicant(s): 日本電気株式会社
 株式会社NEC情報システムズ

2 0 0 4 年 1 月 1 3 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 1 1 0 5 7 7

【書類名】 特許願

【整理番号】 53211036

【提出日】 平成15年 2月18日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/16

【発明者】

 【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

 【氏名】 桑原 賢二

【発明者】

 【住所又は居所】 神奈川県川崎市高津区坂戸 3 丁目 2 番 1 号 株式会社エヌイーシー情報システムズ内

 【氏名】 神田 正己

【特許出願人】

 【識別番号】 000004237

 【氏名又は名称】 日本電気株式会社

【特許出願人】

 【識別番号】 394017491

 【氏名又は名称】 株式会社エヌイーシー情報システムズ

【代理人】

 【識別番号】 100077827

 【弁理士】

 【氏名又は名称】 鈴木 弘男

【手数料の表示】

 【予納台帳番号】 015440

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

【包括委任状番号】 9303403

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法

【特許請求の範囲】

【請求項 1】 単一メモリ空間上で走行するソフトウェアをマルチプロセッサ上で動作させるための実行形式プログラムを生成する際、オブジェクトファイル単位で、各プロセッサへ割り当て、それぞれのプロセッサ毎に実行形式プログラムを生成することを特徴とする単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法。

【請求項 2】 各プロセッサが管理するメモリ空間に実装される前記実行形式プログラムは、プロセッサ間でアドレスが重複しないように配置されることを特徴とする請求項 1 に記載の単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法。

【請求項 3】 前記実行形式プログラムが走行中に、他のプロセッサが管理するメモリ空間上に配置された変数への参照が発生した場合、参照元のプロセッサが持つ例外処理を起動させることにより当該事象を検出し、適切なプロセッサへ参照要求し、参照要求されたプロセッサは変数を参照し、参照結果を参照元に返信し、参照元のプロセッサは、返信結果から変数参照命令をエミュレーション実行し、例外処理から次ぎの命令へ復帰することを特徴とする請求項 1 または 2 に記載の単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法。

【請求項 4】 前記実行形式プログラムが走行中に、他のプロセッサが管理するメモリ空間上に配置された変数への書きこみが発生した場合、書きこみ要求元のプロセッサが持つ例外処理を起動させることにより当該事象を検出し、適切なプロセッサへ書きこみ要求し、書きこみ要求されたプロセッサは変数に書きこみし、書きこみ要求元のプロセッサは、例外処理から次ぎの命令へ復帰することを特徴とする請求項 1 または 2 に記載の単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法。

【請求項 5】 前記実行形式プログラムが走行中に、他のプロセッサが管理

するメモリ空間上に配置された関数の呼び出しが発生した場合、呼び出し要求元のプロセッサが持つ例外処理を起動させることにより当該事象を検出し、適切なプロセッサへ呼び出し要求し、呼び出し要求されたプロセッサは関数を呼び出し、呼び出し結果を呼び出し要求元に返信し、呼び出し要求元のプロセッサは、返信結果から関数呼び出し命令をエミュレーション実行し、例外処理から次ぎの命令へ復帰することを特徴とする請求項 1 または 2 に記載の単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法。

【請求項 6】 前記例外処理を介しての処理要求送信と該処理結果返信の通信を可能とするプロセッサ間通信を備えていることを特徴とする請求項 3、4 または 5 に記載の単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法。

【請求項 7】 請求項 1 ないし 6 のうちのいずれか 1 項に記載の変換する方法によって変換されたマルチプロセッサ用ソフトウェアプログラムを実装したことを特徴とする携帯電話機。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法に関する。

【0 0 0 2】

【従来の技術】

通常、ソフトウェアプログラム（以下、単に「プログラム」ともいう）を開発する際には、エディタ等によってソースファイルを書き上げ、それをコンパイラによってコンパイルしてオブジェクトファイルを生成し、さらにオブジェクトファイルをリンカに通すことによってプログラム内の関数や変数等のアドレス割り当て、リンクを行い、実行形式ファイルすなわち実行形式プログラムを生成する。

【0 0 0 3】

このようにして作成されるソフトウェアプログラムのソースファイルは、その動作環境たとえばハードウェアの構成に合わせて作成される。したがって、単一プロセッサ用ソフトウェアプログラムは、単一プロセッサであるというハードウェア構成を前提にして作成される。

【0 0 0 4】

ところが最近では、ハードウェアコストの低減を背景に、単一プロセッサで構成された装置をマルチプロセッサにすることによる処理能力の向上が望まれてきており、これに対応し、単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用に変換する要求が生じてきている。

【0 0 0 5】

そこで、特許文献 1 では、単一プロセッサ用に作成されたソースファイルに対してプログラム並列化処理を施し、複数のプロセッサを有するハードウェア構成に対応したプログラムのソースファイルを生成するプログラム並列方法を開示している。

【0 0 0 6】

【特許文献 1】

特開平 7 - 1 1 4 5 1 6 号公報

【発明が解決しようとする課題】

しかしながら、従来の方法には次のような問題があった。

【0 0 0 7】

従来一般的には、単一プロセッサ用に作成されたソフトウェアプログラムをマルチプロセッサ用に変換するためには、単一プロセッサ用ソフトウェアプログラムのソースファイルをマルチプロセッサ用に書きなおす必要があり、この作業を行うには、単一プロセッサ用ソフトウェアプログラムと等価な動作を保証しなければならないため、複数のプロセッサ間で独立に処理が可能となるようソフトウェアプログラムの論理を理解した上でソフトウェアプログラムの構造や実行単位を根本から見直し、分割し、論理を再構築を行う必要があった。

【0 0 0 8】

こうなると、作業者は、単一プロセッサ用ソフトウェアプログラムの構造を熟

知するとともに、マルチプロセッサとなったハードウェア構成についても熟知する必要がある、対応できる人が限られてしまうし、変換作業に大変な手間がかかるとともに、複雑な作業のためバグ発生のおそれが増加してしまうという問題があった。

【0009】

また、特許文献1に記載のプログラム並列方法では、単一プロセッサ用ソフトウェアプログラムのソースファイルに基づいてマルチプロセッサ用ソフトウェアプログラムのソースファイルを自動的に生成するようにしており、不必要なプロセッサ間通信を行わないようプログラムを生成するようにしているが、生成されたマルチプロセッサ用ソフトウェアプログラムのソースファイルは、いわば機械的に作成されたものであり、人間がプログラミングした場合と比べると、どうしても冗長的な処理を記載してしまったりして、人間がプログラミングしたものと比べて処理速度や必要なメモリ容量の点などで不利なものになってしまうおそれがあり、元となる単一プロセッサ用ソフトウェアプログラムの長所を消してしまうおそれもあった。

【0010】

本発明は上記の点にかんがみてなされたもので、単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する際に、元となる単一プロセッサ用ソフトウェアプログラムの長所を生かしつつ、変換作業の手間を解消し、バグ発生のおそれも低減することができる、単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法を提供することを目的とする。

【0011】

【課題を解決するための手段】

本発明は上記の目的を達成するために、単一メモリ空間上で走行するソフトウェアをマルチプロセッサ上で動作させるための実行形式プログラムを生成する際、オブジェクトファイル単位で、各プロセッサへ割り当て、それぞれのプロセッサ毎に実行形式プログラムを生成することを特徴とする。

【0012】

また本発明は、各プロセッサが管理するメモリ空間に実装される前記実行形式プログラムは、プロセッサ間でアドレスが重複しないように配置されることを特徴とする請求項 1 に記載の単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法。

【 0 0 1 3 】

また本発明は、前記実行形式プログラムが走行中に、他のプロセッサが管理するメモリ空間上に配置された変数への参照が発生した場合、参照元のプロセッサが持つ例外処理を起動させることにより当該事象を検出し、適切なプロセッサへ参照要求し、参照要求されたプロセッサは変数を参照し、参照結果を参照元に返信し、参照元のプロセッサは、返信結果から変数参照命令をエミュレーション実行し、例外処理から次ぎの命令へ復帰することを特徴とする。

【 0 0 1 4 】

また本発明は、前記実行形式プログラムが走行中に、他のプロセッサが管理するメモリ空間上に配置された変数への書きこみが発生した場合、書きこみ要求元のプロセッサが持つ例外処理を起動させることにより当該事象を検出し、適切なプロセッサへ書きこみ要求し、書きこみ要求されたプロセッサは変数に書きこみし、書きこみ要求元のプロセッサは、例外処理から次ぎの命令へ復帰することを特徴とする。

【 0 0 1 5 】

また本発明は、前記実行形式プログラムが走行中に、他のプロセッサが管理するメモリ空間上に配置された関数の呼び出しが発生した場合、呼び出し要求元のプロセッサが持つ例外処理を起動させることにより当該事象を検出し、適切なプロセッサへ呼び出し要求し、呼び出し要求されたプロセッサは関数を呼び出し、呼び出し結果を呼び出し要求元に返信し、呼び出し要求元のプロセッサは、返信結果から関数呼び出し命令をエミュレーション実行し、例外処理から次ぎの命令へ復帰することを特徴とする。

【 0 0 1 6 】

また本発明は、前記例外処理を介しての処理要求送信と該処理結果返信の通信を可能とするプロセッサ間通信を備えていることを特徴とする。

【0017】**【発明の実施の形態】**

次に本発明の実施の形態について図面を参照して詳細に説明する。

【0018】

図1は、本発明による単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法の一実施の形態を実行する装置の一例であるプログラム開発用端末装置の概要を示す外観図である。

【0019】

本実施の形態において、単一プロセッサ用ソフトウェアプログラムからマルチプロセッサ用ソフトウェアプログラムへの変換は、図1に示すプログラム開発用端末装置10で行われる。

【0020】

図1に示すように、プログラム開発用端末装置10は、たとえばパーソナルコンピュータを用いることができ、マウスやキーボード等の入力手段13およびディスプレイ等の表示手段14を備えて構成される。

【0021】

図2は、図1に示したプログラム開発用端末装置10の内部構成を示すブロック図である。

【0022】

プログラム開発用端末装置10は、図2に示すように、全体の動作を制御するCPU11と、CPU11で動作するアプリケーションファイルや各種データ等を記憶する記憶手段12と、マウスやキーボード等の入力手段13と、ディスプレイ等の表示手段14とを備えて構成される。

【0023】

また、記憶手段12には、CPU11で動作するアプリケーションファイルの一種である、コンパイラ2およびリンカ4が記憶されている。このコンパイラ2およびリンカ4は、単一プロセッサ用ソフトウェアプログラムやマルチプロセッサ用ソフトウェアプログラムの生成に用いられる。

【0024】

図3は、本発明による単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法の一実施の形態によって生成された、マルチプロセッサ用ソフトウェアプログラムによって動作する装置の一例である携帯電話機の概要を示す外観図である。

【0025】

本実施の形態では、プロセッサエレメント（以下「PE」という）が単一の携帯電話機を動作させるための単一プロセッサ用ソフトウェアプログラムを、PEが2つの携帯電話機を動作させるためのマルチプロセッサ用ソフトウェアプログラムに変換する例について説明する。

【0026】

図3に示すように、携帯電話機20は、アンテナ23と、プッシュ式のボタンスイッチ等を備えた入力手段39と、LCD等を備えた表示手段36と、マイク等を備えた送話手段37と、スピーカ等を備えた受話手段38とを備えて構成される。

【0027】

図4は、図3に示した携帯電話機20の内部構成を示すブロック図である。

【0028】

携帯電話機20は、図4に示すように、全体の動作を制御する制御手段21と、通話等のため無線の送受信をする送受信手段22およびアンテナ23と、プッシュ式のボタンスイッチ等を備えた入力手段39と、LCD等を備えた表示手段36と、マイク等を備えた送話手段37と、スピーカ等を備えた受話手段38とを備えて構成される。

【0029】

また、本実施の形態において、制御手段21は、2つのプロセッサエレメントであるPE26とPE31とを有する。本発明はこれに限られるものではなく、3つ以上のPEを有する装置にも適用することができるというまでもない。

【0030】

PE26は、CPU27と、PE26とPE31との通信を行うPE間通信手段28と、CPU27で動作するプログラム等を実行時に展開するメモリ29と

、CPU27で動作するプログラムや各種データ等を不揮発的に記憶する不揮発性メモリ30とを有して構成される。

【0031】

PE31は、CPU32と、PE31とPE26との通信を行うPE間通信手段33と、CPU32で動作するプログラム等を実行時に展開するメモリ34と、CPU32で動作するプログラムや各種データ等を不揮発的に記憶する不揮発性メモリ35とを有して構成される。

【0032】

図5は、単一プロセッサ用ソフトウェアプログラムの実行形式ファイルを生成する手順を説明する図である。

【0033】

本実施の形態においては、図5に示すように、ソースファイル1a～1eの5個のソースファイルから、単一プロセッサ用ソフトウェアプログラムである1つの実行形式ファイル（実行形式プログラム）5を生成する場合について説明する。

【0034】

まずソースファイル1a～1eのそれぞれを、コンパイラ2によってコンパイルし、これによってオブジェクトファイル3a～3eを生成する。また、オブジェクトファイル3a～3eのすべてをリンカ4によってリンクし、単一プロセッサ用ソフトウェアプログラムである1つの実行形式ファイル5を生成する。

【0035】

続いて、図5に示したソースファイル1a～1eから、2つのPEに対応したマルチプロセッサ用ソフトウェアプログラムである2つの実行形式ファイルを生成する場合について説明する。

【0036】

図6は、マルチプロセッサ用ソフトウェアプログラムの実行形式ファイルを生成する手順を説明する図である。

【0037】

本実施の形態においては、図6に示すように、図5に示したのと同じソースフ

ファイル 1 a ～ 1 e の 5 個のソースファイルから、マルチプロセッサ用ソフトウェアプログラムである 2 つの実行形式ファイル（実行形式プログラム） 6 a、6 b を生成する場合について説明する。

【 0 0 3 8 】

まずソースファイル 1 a ～ 1 e のそれぞれを、コンパイラ 2 によってコンパイルし、これによってオブジェクトファイル 3 a ～ 3 e を生成する。

【 0 0 3 9 】

次に、オブジェクトファイル 3 a ～ 3 e を P E の数（本実施の形態では図 4 に示したように P E の数は 2 つ）のグループに分け、グループごとにリンカ 4 によってリンクする。図 6 に示した例では、オブジェクトファイル 3 a ～ 3 c をリンクすることによって実行形式ファイル 6 a を生成し、オブジェクトファイル 3 d ～ 3 e をリンクすることによって実行形式ファイル 6 b を生成する。

【 0 0 4 0 】

ここで、たとえばオブジェクトファイル 3 a に含まれる関数が、自分と別のグループのオブジェクトファイルである 3 d に実体が存在する関数や変数を参照している場合には、オブジェクトファイル 3 a と 3 b と 3 d とでリンクをする際に実体がない関数や変数のアドレスがわからず、リンクできない状態となってしまう。そこで、本実施の形態では、リンクの際に、関数や変数の実体が存在するグループでアドレスが確定したならば、その確定したアドレスを実体がなく参照しているグループのリンクの際に通知してやり、これによってリンクを完成させる。

【 0 0 4 1 】

このようにして完成した実行形式ファイル 6 a は図 4 に示した P E 2 6 の不揮発性メモリ 3 0 に記憶され、メモリ 2 9 に展開されて C P U 2 7 によって実行される。また、実行形式ファイル 6 b は図 4 に示した P E 3 1 の不揮発性メモリ 3 5 に記憶され、メモリ 3 4 に展開されて C P U 3 2 によって実行される。

【 0 0 4 2 】

図 7 は、本実施の形態によって生成されたマルチプロセッサ用ソフトウェアプログラムが割り当てられるアドレス空間を示す図であり、（a）は図 4 に示した

PE 26 のメモリ 29 に実体があるメモリ空間を示す図であり、(b) は図 4 に示した PE 31 のメモリ 34 に実体があるメモリ空間を示す図であり、(c) は図 4 に示した PE 26 の CPU 27 や PE 31 の CPU 32 が取り扱う仮想的なメモリ空間を示す図である。

【0043】

図 7 (a) ~ 図 7 (c) に示すように、本実施の形態の CPU 27 および CPU 32 はそれぞれが、アドレス 0000h ~ ffffh をプログラム実行の際のメモリ空間としており、PE 26 ではそのうちの 0000h ~ 7ffffh をメモリ 29 に実体として有し、PE 31 ではそのうちの 8000h ~ fffffh をメモリ 34 に実体として有している。

【0044】

図 8 は、図 6 に示した実行形式ファイル 6a が PE 26 のメモリ 29 に展開された状態を示す図である。

【0045】

本実施の形態では、図 8 に示すように、実行形式ファイル 6a は、関数 A、関数 B、関数 C、変数 D、変数 E、変数 F および変数 G の実体を有し、それぞれがメモリ 29 のアドレス空間に展開される。

【0046】

また、メモリ 29 には Data abort 時割込処理格納領域 45 が設けられている。この Data abort 時割込処理格納領域 45 は、CPU 27 がメモリ 29 に実体のアドレスがない変数の読み出し（参照）や書きこみを行おうとしたときに、いわゆる例外処理の一種である、割込みで実行する処理を格納する領域である。

【0047】

Data abort 時割込処理格納領域 45 に格納される割込み処理は、割り込みの原因となった、メモリ 29 に実体のアドレスがない変数の読み出しや書きこみを、PE 間通信手段 28 を使用して PE 31 と通信し、変数の実体があるメモリ 34 に対して行うものである。このため、PE 31 にはこれに呼応して実際にメモリ 34 に対してその変数のアクセスを行うプログラムが必要となる。

【0048】

さらに、メモリ29にはPrefetch abort時割込処理格納領域46が設けられている。このPrefetch abort時割込処理格納領域46は、CPU27がメモリ29に実体のアドレスがない関数の呼び出しを行おうとしたときに、いわゆる例外処理の一種である、割込みで実行する処理を格納する領域である。

【0049】

Prefetch abort時割込処理格納領域46に格納される割込み処理は、割り込みの原因となった、メモリ29に実体のアドレスがない関数の呼び出しを、PE間通信手段28を使用してPE31と通信し、関数の実体があるメモリ34に対して行うものである。このため、PE31にはこれに呼応して実際にメモリ34に対してその関数の呼び出しを行い、実行するプログラムが必要となる。

【0050】

なお、Data abort時割込処理格納領域45やPrefetch abort時割込処理格納領域46に格納される割込み処理は予めプログラミングして不揮発性メモリ30に記憶しておき、実行時にメモリ29に展開するようにしてもよい。また、Data abort時割込処理格納領域45やPrefetch abort時割込処理格納領域46はCPU27内に設けてもよい。

【0051】

図9は、図6に示した実行形式ファイル6bがPE31のメモリ34に展開された状態を示す図である。

【0052】

本実施の形態では、図9に示すように、実行形式ファイル6bは、関数J、関数K、変数L、変数M、変数N、変数Oおよび変数Pの実体を有し、それぞれがメモリ34のアドレス空間に展開される。

【0053】

また、メモリ34にはData abort時割込処理格納領域50が設けられている。このData abort時割込処理格納領域50は、CPU32が

メモリ 3 4 に実体のアドレスがない変数の読み出しや書きこみを行おうとしたときに、いわゆる例外処理の一種である、割込みで実行する処理を格納する領域である。

【 0 0 5 4 】

D a t a a b o r t 時割込処理格納領域 5 0 に格納される割込み処理は、割り込みの原因となった、メモリ 3 4 に実体のアドレスがない変数の読み出しや書きこみを、P E 間通信手段 3 3 を使用して P E 2 6 と通信し、変数の実体があるメモリ 2 9 に対して行うものである。このため、P E 2 6 にはこれに呼応して実際にメモリ 2 9 に対してその変数のアクセスを行うプログラムが必要となる。

【 0 0 5 5 】

さらに、メモリ 3 4 には P r e f e t c h a b o r t 時割込処理格納領域 5 1 が設けられている。この P r e f e t c h a b o r t 時割込処理格納領域 5 1 は、C P U 3 2 がメモリ 3 4 に実体のアドレスがない関数の呼び出しを行おうとしたときに、いわゆる例外処理の一種である、割込みで実行する処理を格納する領域である。

【 0 0 5 6 】

P r e f e t c h a b o r t 時割込処理格納領域 5 1 に格納される割込み処理は、割り込みの原因となった、メモリ 3 4 に実体のアドレスがない関数の呼び出しを、P E 間通信手段 3 3 を使用して P E 2 6 と通信し、関数の実体があるメモリ 2 9 に対して行うものである。このため、P E 2 6 にはこれに呼応して実際にメモリ 2 9 に対してその関数の呼び出しを行い、実行するプログラムが必要となる。

【 0 0 5 7 】

なお、D a t a a b o r t 時割込処理格納領域 5 0 や P r e f e t c h a b o r t 時割込処理格納領域 5 1 に格納される割込み処理は予めプログラミングして不揮発性メモリ 3 5 に記憶しておき、実行時にメモリ 3 4 に展開するようにしてもよい。また、D a t a a b o r t 時割込処理格納領域 4 5 や P r e f e t c h a b o r t 時割込処理格納領域 4 6 は C P U 2 7 内に設けてもよい。

【 0 0 5 8 】

ここで、上記の Data abort 時割込処理格納領域 45 や Prefetch abort 時割込処理格納領域 46 や Data abort 時割込処理格納領域 50 や Prefetch abort 時割込処理格納領域 51 に格納される割込み処理である abort 時処理についてさらに説明する。

【0059】

図 10 は、本実施の形態における abort 時処理のフローチャートを示す図である。

【0060】

各 PE では、実体が自分側でないアドレスに対するアクセス（データの読み出し、書きこみ、関数の呼び出し）要求があった場合、割込み処理が実行される。この割込み処理では、PE 間通信を行い、自分以外の PE に対してアクセス要求の実行を依頼する（A-1）。依頼を受けた PE ではその実体が自身にあればその依頼を実行し、必要に応じて実行結果を依頼元の PE に返す（A-2）。依頼元の PE では割込み処理の完了により、割込み前の処理に復帰する（A-3）。

【0061】

なお、上述の実施の形態では、携帯電話機用の単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法について説明したが、本発明が携帯電話機用に限定されるものでないことはいうまでもない。

【0062】

以上説明したように本発明によれば、単一メモリ空間上で走行するソフトウェアをマルチプロセッサ上で動作させる場合、ソフト構造や論理を変更することなく、ソースファイル、オブジェクトファイル、ライブラリ等のソフトウェアの構成要素単位でプロセッサへの割り付けを考えればよいので、開発や検証の期間、およびそれら作業に要する費用を抑えることが可能となる。

【0063】

【発明の効果】

以上説明したように、本発明によれば、単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する際に、元となる単

一プロセッサ用ソフトウェアプログラムの長所を生かしつつ、変換作業の手間を解消し、バグ発生のおそれも低減することができる、単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法を提供することができる。

【 0 0 6 4 】

すなわち、本発明によれば、単一メモリ空間上で走行するソフトウェアをマルチプロセッサ上で動作させる場合、ソフト構造や論理を変更することなく、ソースファイル、オブジェクトファイル、ライブラリ等のソフトウェアの構成要素単位でプロセッサへの割り付けを考えればよいので、開発や検証の期間、およびそれら作業に要する費用を抑えることが可能となる。

【図面の簡単な説明】

【図 1】

本発明による単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法の一実施の形態を実行する装置の一例であるプログラム開発用端末装置の概要を示す外観図である。

【図 2】

図 1 に示したプログラム開発用端末装置 1 0 の内部構成を示すブロック図である。

【図 3】

本発明による単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法の一実施の形態によって生成された、マルチプロセッサ用ソフトウェアプログラムによって動作する装置の一例である携帯電話機の概要を示す外観図である。

【図 4】

図 3 に示した携帯電話機 2 0 の内部構成を示すブロック図である。

【図 5】

単一プロセッサ用ソフトウェアプログラムの実行形式ファイルを生成する手順を説明する図である。

【図 6】

マルチプロセッサ用ソフトウェアプログラムの実行形式ファイルを生成する手順を説明する図である。

【図 7】

本実施の形態によって生成されたマルチプロセッサ用ソフトウェアプログラムが割り当てられるアドレス空間を示す図であり、(a)は図4に示したPE26のメモリ29に実体があるメモリ空間を示す図であり、(b)は図4に示したPE31のメモリ34に実体があるメモリ空間を示す図であり、(c)は図4に示したPE26のCPU27やPE31のCPU32が取り扱う仮想的なメモリ空間を示す図である。

【図 8】

図6に示した実行形式ファイル6aがPE26のメモリ29に展開された状態を示す図である。

【図 9】

図6に示した実行形式ファイル6bがPE31のメモリ34に展開された状態を示す図である。

【図 10】

本実施の形態におけるabort時処理のフローチャートを示す図である。

【符号の説明】

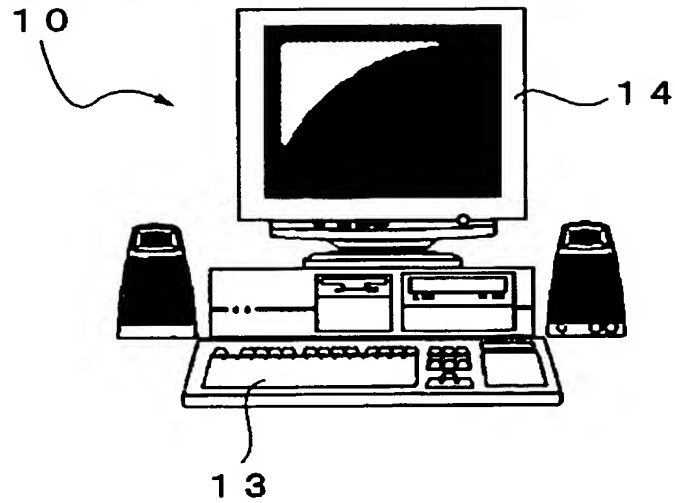
- 1a～1e ソースファイル
- 2 コンパイラ
- 3a～3e オブジェクトファイル
- 4 リンカ
- 5、6a、6b 実行形式ファイル
- 10 プログラム開発用端末装置
- 11 CPU
- 12 記憶手段
- 13 入力手段
- 14 表示手段
- 20 携帯電話機

- 2 1 制御手段
- 2 2 送受信手段
- 2 3 アンテナ
- 2 6、3 1 P E
- 2 7、3 2 C P U
- 2 8、3 3 P E 間通信手段
- 2 9、3 4 メモリ
- 3 0、3 5 不揮発性メモリ

【書類名】 図面

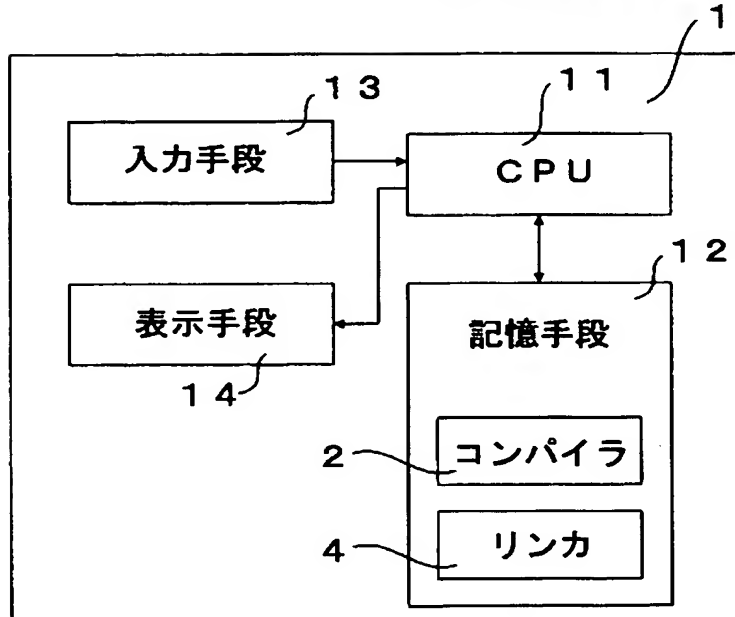
【図 1】

プログラム開発用端末装置

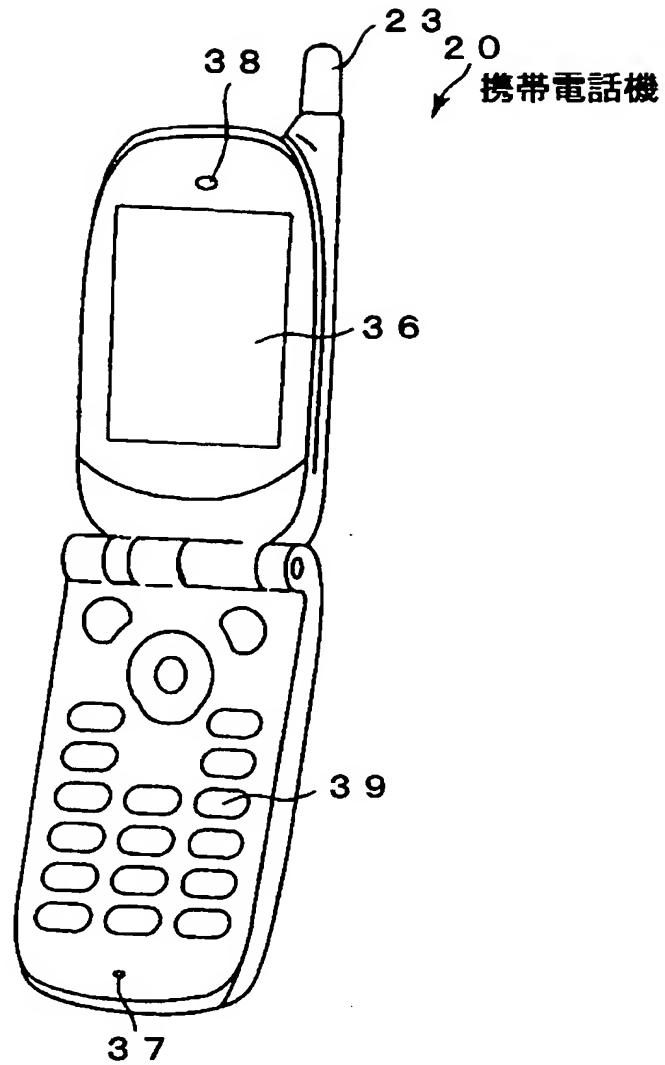


【図 2】

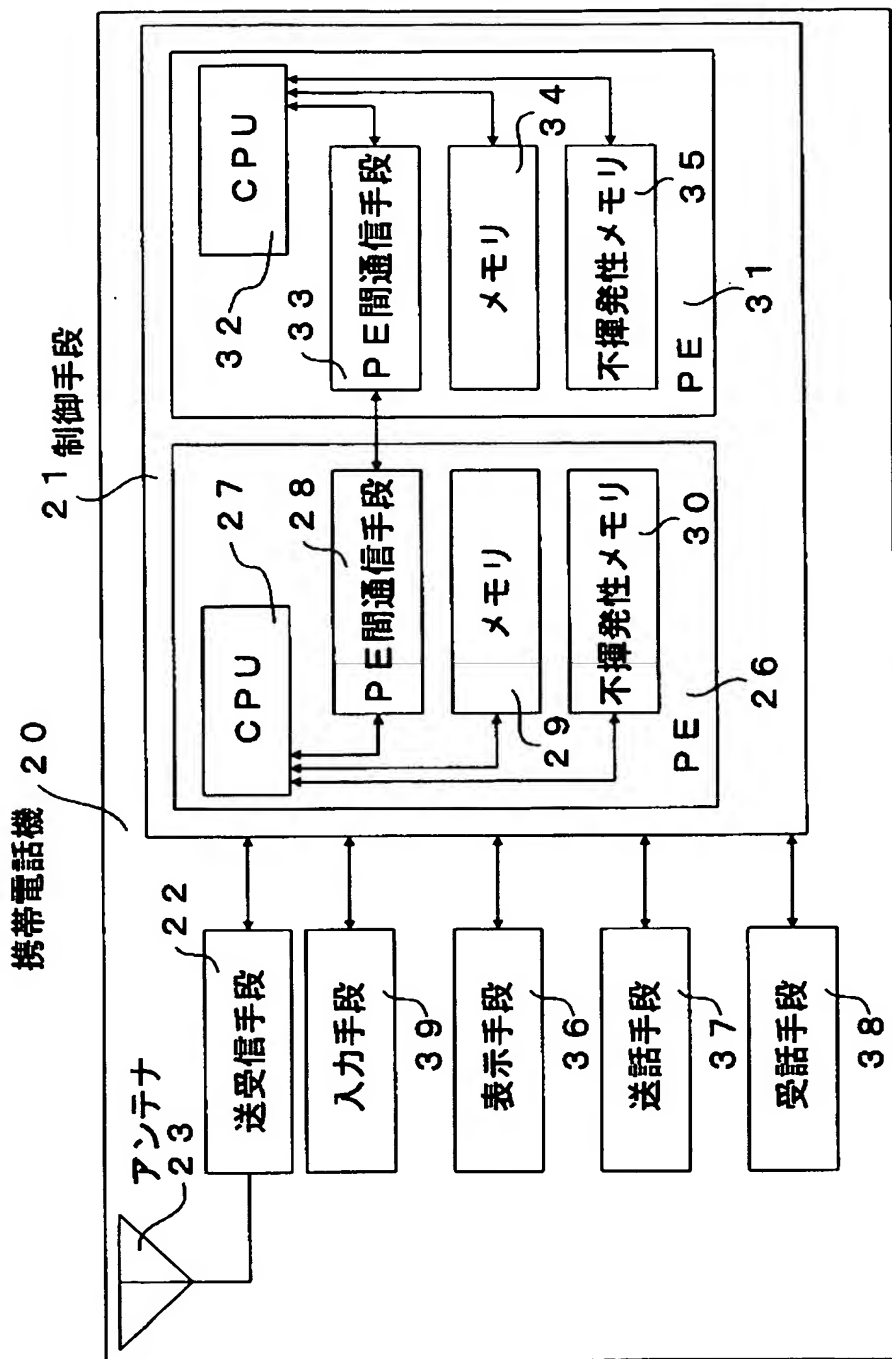
プログラム開発用端末装置



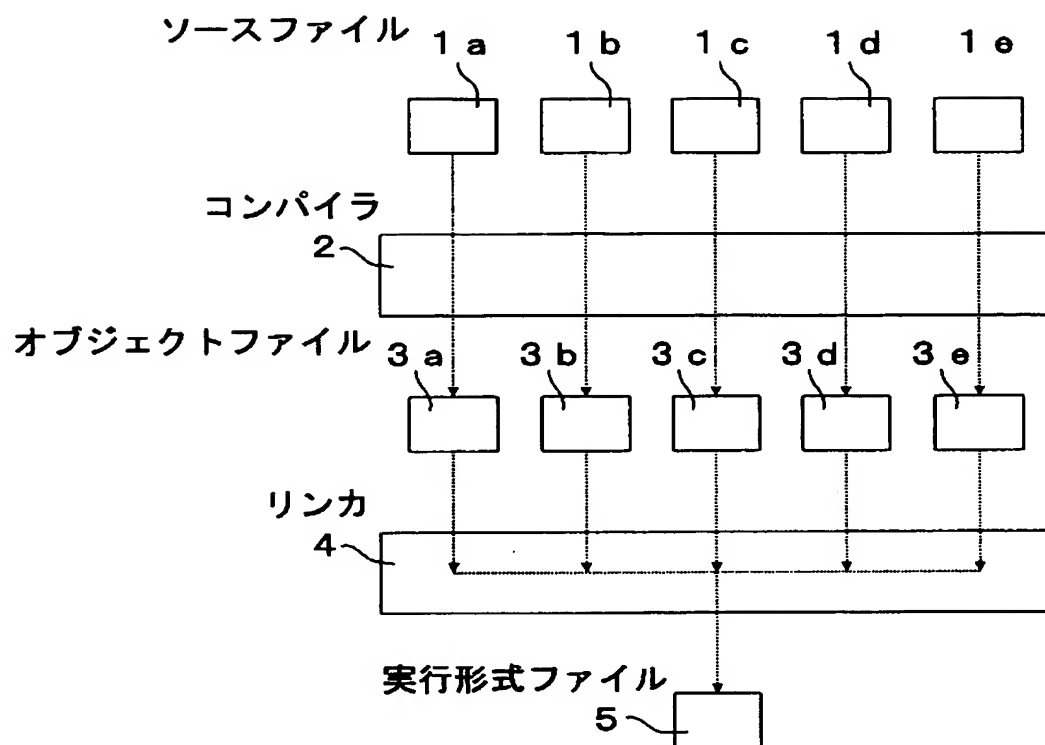
【図 3】



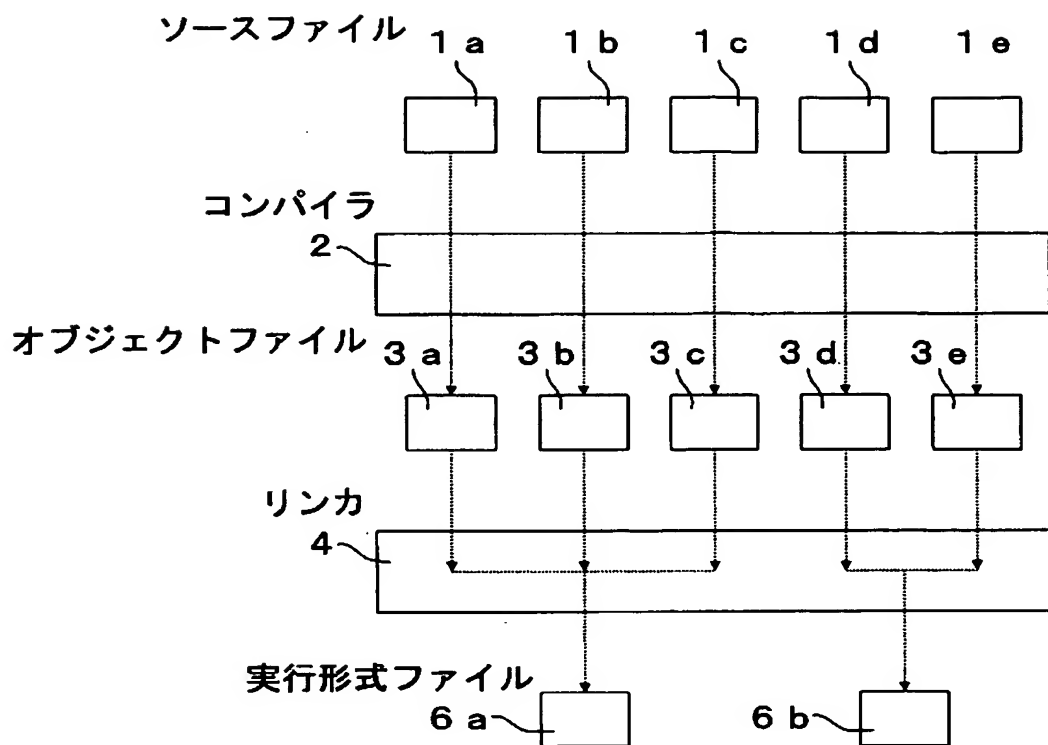
【図 4】



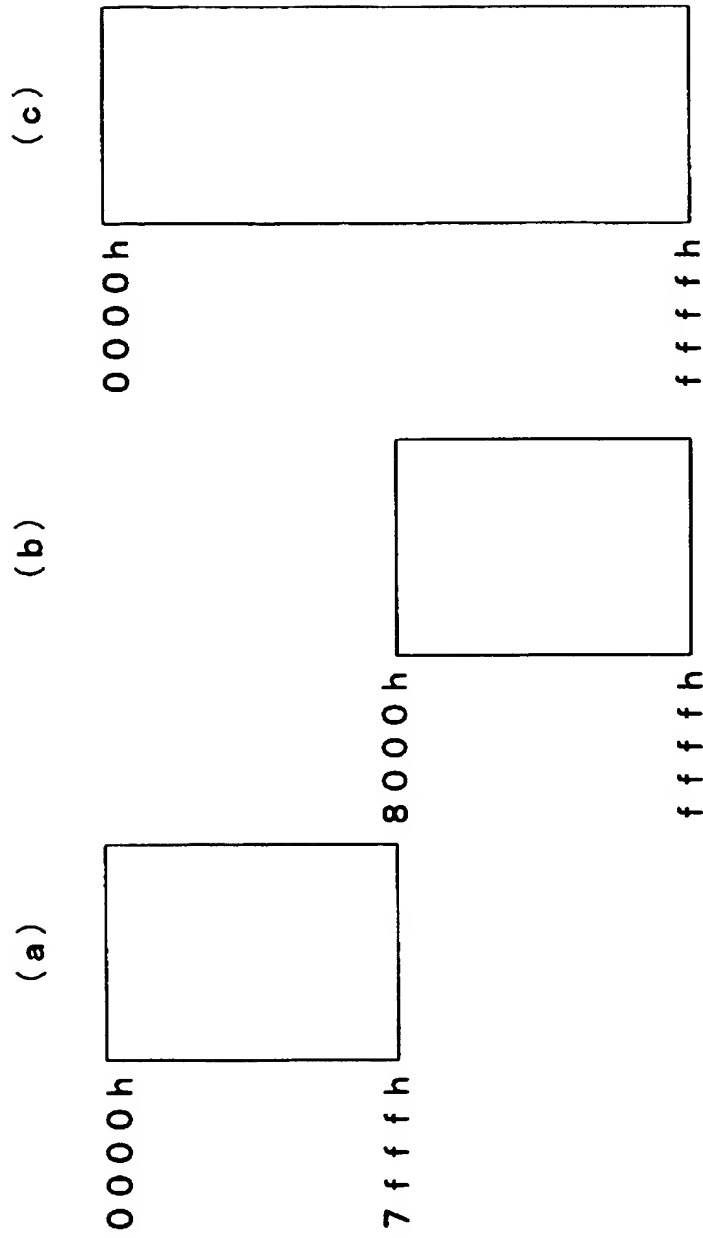
【図 5】



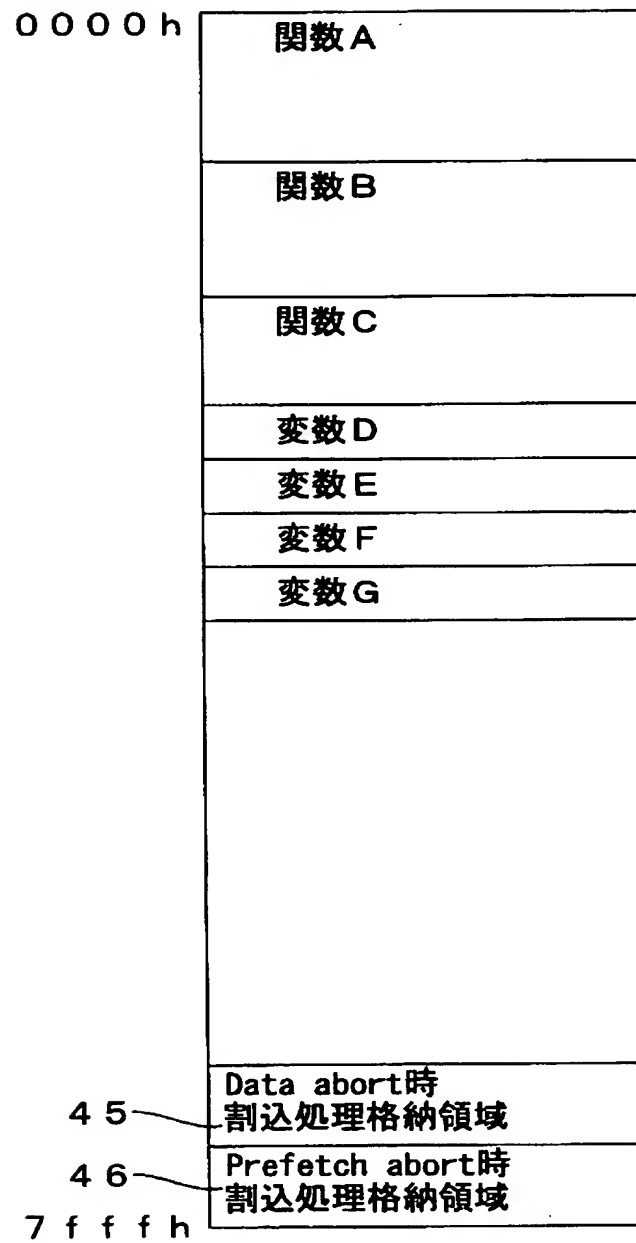
【図 6】



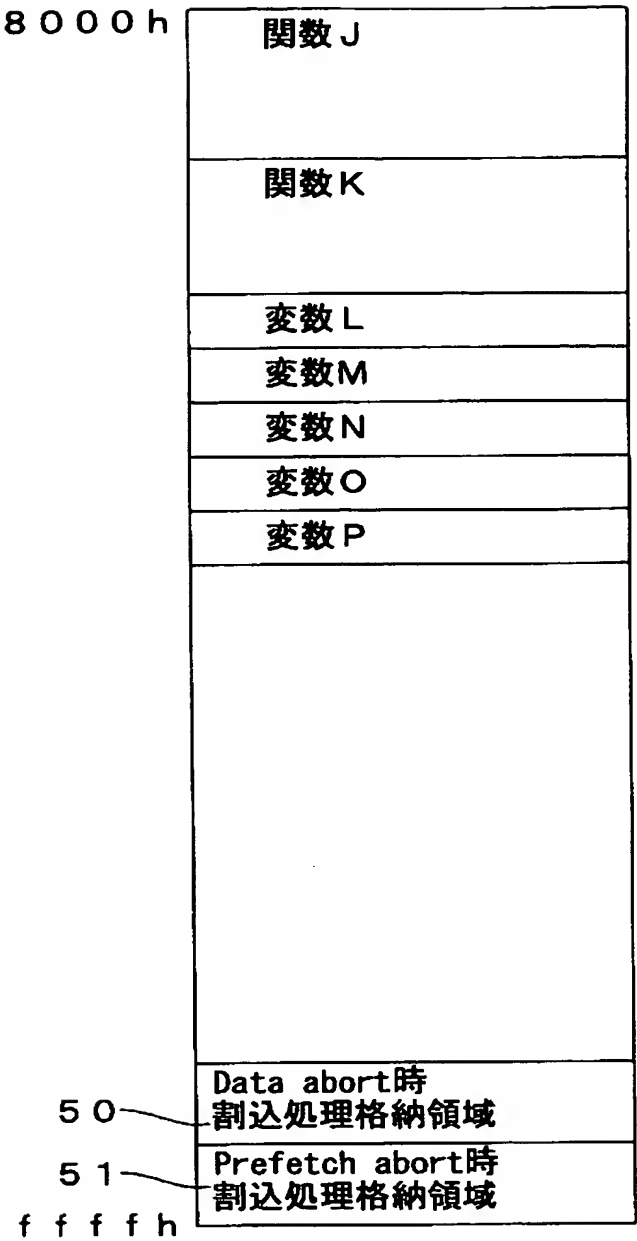
【図 7】



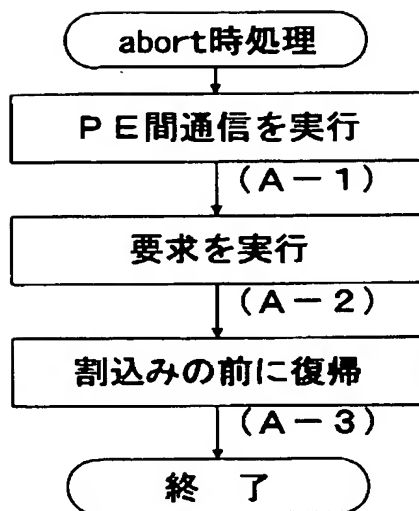
【図 8】



【図 9】



【図 10】



【書類名】 要約書

【要約】

【課題】 単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する際に、元となる単一プロセッサ用ソフトウェアプログラムの長所を生かしつつ、変換作業の手間を解消し、バグ発生のおそれも低減することができる、単一プロセッサ用ソフトウェアプログラムをマルチプロセッサ用ソフトウェアプログラムに変換する方法を提供することである。

【解決手段】 単一メモリ空間上で走行するソフトウェアをマルチプロセッサ上で動作させるための実行形式プログラムを生成する際、オブジェクトファイル単位で、各プロセッサへ割り当て、それぞれのプロセッサ毎に実行形式プログラムを生成することを特徴とする。

【選択図】 図 6

特願 2 0 0 3 - 0 3 9 0 5 9

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 4 2 3 7]

1. 変更年月日	1 9 9 0 年 8 月 2 9 日
[変更理由]	新規登録
住 所	東京都港区芝五丁目 7 番 1 号
氏 名	日本電気株式会社

特願 2 0 0 3 - 0 3 9 0 5 9

出 願 人 履 歴 情 報

識別番号

[3 9 4 0 1 7 4 9 1]

- | | |
|----------|---------------------------|
| 1. 変更年月日 | 1 9 9 6 年 4 月 1 8 日 |
| [変更理由] | 住所変更 |
| 住 所 | 神奈川県川崎市高津区坂戸 3 丁目 2 番 1 号 |
| 氏 名 | 株式会社エヌイーシー情報システムズ |
| | |
| 2. 変更年月日 | 2 0 0 3 年 8 月 2 9 日 |
| [変更理由] | 名称変更 |
| 住 所 | 神奈川県川崎市高津区坂戸 3 丁目 2 番 1 号 |
| 氏 名 | 株式会社 N E C 情報システムズ |